June, 1991
VOL 8, NO. 6
8th Year
¥300

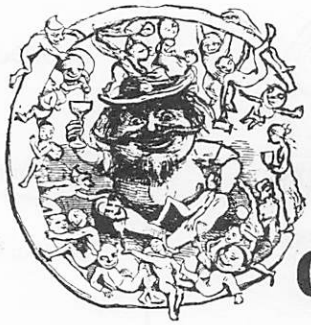
RESUME.TPC

Tokyo

WYSIWYG 1976

ersonal

TPC Club begins A.D. 1981



omputer

Typical TPC Meeting, ca. 1986

Newsletter

WYSIWYG 1991

Largest Personal Computer Users Group in the Greater Asia IBM Coprocessing Sphere

Tokyo PC Newsletter
Published by the Tokyo PC Users' Club
Central P.O Box 1145, Tokyo 100-91, Japan

Purveyors of Fine Hardcopy
Since 1984
to
The Karaoke
Cyberpunks High On DOS

TPC BBS 24 hours a day
300/1200/2400/9600 HST MNP5— 8/N/1
(03) 3997-3444

Information hotline for prospective members:
(03) 5706-0057

The club meeting is held on the first Thursday of each month at the Tokyo Union Church between Omote-Sando Station and Harajuku Station, starting from 6 p.m.

TPC Executive Board

SUE WINSKI, *President*	(03)3714-4448
LARRY CISAR, *Vice President*	(0489)77-5719
TONY WHITMAN, *Treasurer*	(03)3349-8521
MARTIN BRUCZKOWSKI, *Sysop*	(03)3997-7377
STUART WOODWARD, *Programs Director*	(03)5702-1345
MIKE GRAY, *Newsletter Publisher*	(03)3464-8736
MARY JANE LIDDICOAT, *Assistant Publisher*	(03)3806-3079
CHRISTOPHER CASE, *Newsletter Editor*	(0429)78-0791
DAVID BERNAT, *Assistant Editor*	(03)5371-3488
JOSEPH KAYE, *Disk Librarian*	(03)3401-8278
BRUCE PIERCE, *Membership*	(03)5567-1745
ROGER BOISVERT, *President Emeritus*	(03)3581-5571

!!

COMING SOON TO YOUR LOCAL TPC MEETING HALL

JUNE
Computer Music. A Demonstration of MIDI in action.

JULY
Graphics. Hopefully VGA graphic programs can be demonstrated.

AUGUST
User Demonstrations. Everyone is invited to bring in their favourite software and take the floor.

SEPTEMBER
Programming. Getting Started. Batch Files. Compiled Batchfiles, C++.

OCTOBER
DOS 5.0. Dependent on the release date, but be prepared for the biggest crowd you have ever seen.

NOVEMBER
Word Processing, Desktop Publishing and Translating tools.

DECEMBER
Entertainment: Games, simulations, and other leisure activities.

JANUARY
Business Software.

FEBRUARY
Special Interest Groups.

MARCH
Hardware.

APRIL
Communications.

Subscribe!

One year's mailed subscription to this magazine costs ¥6,000. Full membership of the Tokyo PC Users' Club, which includes the subscription and full use of the Bulletin Board, costs ¥10,000 per year plus an additional ¥10,000 at the start. Send a check or postal money order for the appropriate figure to the address on page 2. For further details (e.g. of corporate, family, or student membership options), please write, call the hotline, or ask on the BBS.
Please address correspondence concerning the Newsletter to: The Editor, TPC Newsletter, Hirato 82-1, Hanno-shi, Saitama-ken, Japan 357-02 TEL:0429-78-0791 FAX:0429-78-0681 E-Mail: TPC BBS User #161

SEZ THE PREZ

To Join Or Not To Join

There are people who come regularly to the meetings but do not join the club. "I can pay 1,000 yen entrance fee for 12 months, less than the first year's membership, and even cheaper if I don't go to all the meetings. Why should I join?" they ask.

If somebody comes to the meetings and benefits from them, then he/she should join. TPC continues to exist because of its members. TPC is more than the meetings, the newsletter, the BBS. At the most basic level, it is a unique network of people. Members enable us to continue our existence. It is very simple. No members, no TPC. Those who keep coming without joining are parasites. Some people say they don't get their money's worth. I say, that's partly THEIR fault. Those who participate get their money's worth. TPC provides a structure to make things happen.

The executive members and others as well are volunteering their time, time which, like land, is probably more valuable in Tokyo than anywhere else. Nobody wants to spend this kind of time servicing freeloaders. Invective over, I feel better now.

As a postscript: if anyone is truly having financial problems, talk to me, talk to Tony (treasurer) or Bruce (membership). Perhaps we can work something out.

Your comments are welcome, about our dues structure, or anything actually.

Sue Winski

SYSOPERA

Hello Users. You know what the BBS is. In a moment you're going to hear: The *Rest* of The Menu.

While trying to work as my wife waltzes with Chopin downstairs, I often catch a glimpse of your logon. What do I see? I'll tell you what I don't see: anybody using all the wonderful features that the Creator (Wayne Bell) hath cast upon us. Oh—sorry. I've forgotten that YOU are the one exceptional BBS user who proves the rule. But others...

Last month, co-sysop Stuart acquainted you with some of the basic commands of WWIV. This time, I wish to list three of less known and even less used commands of WWIV.

"Z"—The notoriously ignored **Z)ippy Message Scan**. I see many users happily browsing the message boards as life rushes past. Fewer are those who type "N" after checking their mail. Those who have discovered the wonders of the **"Z"** command I could count on the fingers of one hand (well, maybe 1.5.). The **"Z"** command is very much like **N)ew**; it displays all messages posted since your last logon. The important difference is that the **Z)ippy scan** doesn't pause between messages, so it's impossible for you to read on-line. The idea behind the **"Z"** command is to capture all unread messages onto your hard drive to read offline. Virtually all communication programmes have a "Log On/Off" or a "Screen Capture" feature. Turn it on, do a "Z" scan, and log off. Now you have all the time in the world to read them.

"I"—System News and I)nfo. This command shows current events. I have been asked several times how to redisplay this info, which speeds down your screen just after the logon and just before you turn on your screen capture. The current news message contains many useful reminders: Club meetings, Paul McCartney's birthday, and the date of the next Harumi Data Show. I continually update those lines, so why not check them out?

"CTRL-T"—This command displays today's **T)ime and date**, and, even more importantly, it shows you how long you have been on line and how much time remains in your session.

And *now* you know—the *rest* of the menu.

Martin Bruczkowski

DISK O' the MONTH

CSHOW is now up to version 8.21. The latest version of this graphics viewer can also display text. It supports just about every major graphics format apart from TIFF. *CSHOW* is still missing a few major features that would be useful—recognition of drive prompts is still not available without pressing the {F2} key. I would also like to be able to type *.GIF and have the program accept it as the file mask I want to display without my pressing {F4}. Additions like a built-in slideshow and mouse support would be nice.

QEdit is a very popular text editor, good for replacing EDLIN for modifying source code or batch files, or even for writing letters. Many people do use it and recommend it: Maynard Hogg has often called it his favourite piece of software. Why use an editor over an ordinary word cruncher? Because, unless you use a RAMDisk or a very fast hard-disk-caching system, loading can be time-consuming. Try *QEdit* now and then for a few weeks, and see if it benefits you.

Well, that's the actual **Disk of the Month.** But there also are **Permanent Disks**. ***VGA Utilities***, a collection of VGA movies, and *Jungle Jim, Telix, Archiver, Virus Removal* (aka *More Survival*), *Sounds #1* and *#2: Sounds #1* sold out last meeting, and only one copy was left of *#2*. Depending on the market, there may be a *Sounds #3*. You can play these sounds on a Macintosh, Amiga, or IBM machine as long as you can convert the disks (Apple File Exchange or the Amiga equivalent). Five copies of DoM #007 will also be available for people who missed it last meeting. Finally, a new Permanent Disk: *PC-Write*. This is my first choice for a word processing program. It comes with a spelling checker, macros and shorthand, help system, is compatible with *WordStar* commands (and will automatically convert files), outputs pure ASCII, supports around 700 printers, has mouse support, and can be used for DTP—although for graphics you'll need *Hijaak* or the equivalent. The cost to you? ¥500. If you don't like it, then don't register, stop using it, and you're out only ¥500. If you like it, $99 gets you full registration—and if someone registers with a copy of your program (you get a special serial number), you receive a commission of $25. This is how shareware should be run.

Joseph Kaye

Telecommunications begins with inserting a modem in the space between your computer and your phone line. The first time you try to transfer a file in either direction, however, you are usually faced with some decisions to be made on-line. "PROTOCOL?" is an intimidating question in almost any context, but a little coaching from Temporary Protocol Officer **Ed Zawlocki** *should save you embarrassment when the chips are down (up?)..*

EVERYTHING YOU NEVER WANTED TO HAVE TO KNOW ABOUT PROTOCOLS

When you want to get information from a remote source, you must use a modem. They in themselves aren't that difficult to use. The problem lies in the transfer of information, where you must select a specific protocol from among the types listed in your communications program. You must also make sure that the exact same protocol is used on both sides. Protocols have numerous advantages/disadvantages which can baffle users, especially when the names used for the protocols are not consistent. The following guidelines can help you decide which of the more widely available protocols is best for you. Note that your favorite protocol may not always be the best one depending on line conditions and modem used.

Text (or ASCII) protocol is the most basic method used to exchange messages between computers. It's essentially the same as typing the message very fast. It cannot generally transfer files that contain non-ASCII, or binary, codes, such as executable programs, spreadsheets, database files, and typical word processor documents. However, most word processors can produce a plain ASCII, or non-document, file which can be sent this way.

XModem is probably the most commonly used transfer protocol. It is a comparatively simple protocol which is used to transfer only one file at a time. XModem uses 128-byte packets and has built-in error-correction, which can be either a one-byte *CRC* or *Checksum*. [CRC means Cyclic Redundancy Check. Checksum is a process that adds the bits in the packet and then transmits that sum as the last byte, which is rechecked at the receiving end. Both checks will ask for a re-send of the packet if an error is detected.] XModem is generally faster than YModem if you have a noisy line, but slower on a clean line.

YModem is often known as **YModem Batch**. It uses 1024-byte packets, and can transfer one file or groups of files. One advantage is that it sends the number of files and the total size of the transfer first, so that the receiving computer can estimate the transfer time.

YModem G is a variation of YModem that doesn't acknowledge each packet. If an error is detected the file transfer is simply aborted. It should be used only with inherently error-free connections, such as direct-cable, ISDN, or X.25. It may be used with modems that have built-in error-correction, like MNP, PEP, or LAP-B.

1K-XModem is similar to YModem, except it can only transfer one file at a time. Some systems refer to 1K-XModem as YModem.

Kermit is an often-used protocol that seems to have gained favor in the academic community. It can send single files or file groups, using 2000-byte packets and a one byte checksum. It is very versatile and can handle noisy lines well, though it tends to be slower than other protocols.

CompuServe B is used when transferring information to/from the CompuServe Information Service. It supports both single and group file transfers, and uses error-correction. This protocol should probably not be used for systems other than CompuServe.

Super 8K Modem is a relatively new transfer protocol. It can send single or multiple files at one time. It transfers information in 8000 byte blocks and, in the case of a detected error, only resends the last 1000 bytes. Super 8K sends the number of files and file sizes, so the receiving computer can display both this and the approximate transfer time. File sizes are rounded up to the next higher 128 byte sector, like XModem, rather than the next higher 1000 byte sector, like YModem.

ZModem is based on XModem, but incorporates the best ideas from many transfer protocols. It uses a 32-bit (4-byte) CRC to transfer files, which are in packets the size of the entire file, unlike other methods that subdivide the file.

ZModem CRC is five orders of magnitude more accurate than XModem CRC. This extra protection is vital in high-speed applications. However, ZModem retains XModem CRC capability. ZModem sends the file count and total size to the receiving computer in order to estimate transfer time. It also offers crash recovery; that is, the capability to complete an interrupted file transfer without throwing away the portion already received. ZModem's "on the fly" compression can boost throughput on suitable files, but doesn't help much for compressed files, and actually increases overhead by 3% on these! However, it does contain a method to lower this overhead to about 0.5%, which is a small price to pay for the crash recovery, automatic downloads, and the safety of 32-bit CRC error checking.

HyperProtocol is an error-correcting, streaming protocol that is very fast over noisy or clean lines. It can transfer single files or file groups. HyperProtocol can compress files on-the-fly, so the transfer time can be reduced. It maintains its speed despite propagation delays common to long-distance calls, high-speed modems, X.25, and packet-switching networks.

Next month, Ed demystifies the "V.42bis or CCITT" area of confusion.

indows upon indows

by Martin Bruczkowski

"Late yesterday night I saw a light shine from a Window..."
—Billy J. Kramer and the Dakotas/ The Beatles

Windows is the first spot of light at the end of the long and dark tunnel of DOS. How long before we emerge from that tunnel? Ask Microsoft, for no one else has the power to make their environment a world standard. Yes, there are better, faster GUIs than *Windows* (look for a short review of GEOS in this column next month). But so far we seem destined to adopt the brainchild of Microsoft and have little to say about it. Let us then try to make the best of what we have. Hence this month's column: *Windows Tips and Tricks*

Tip #1: Install icons for all the applications Microsoft is trying to hide. When you install *Windows*, it creates icons for all programs that come with it. All, except for:

MSDOS.EXE—MS-DOS Executive was the main *Windows* shell in release 2.xx. It is much simpler and clearer than *File Manager* and you may prefer it for quick file management jobs. You can even make it your primary *Windows* shell, by changing the line: "SHELL=PROGMAN.EXE" in SYSTEM.INI to "SHELL=MSDOS.EXE".

SYSEDIT.EXE—System editor is a God-sent utility for editing AUTOEXEC.BAT, CONFIG.SYS, WIN.INI and SYSTEM.INI. It is just a simple ASCII editor.

SWAPFILE.EXE—SwapFile creates a permanent swap file on the hard disk—an area which *Windows* can use as virtual memory on 386 machines. *SwapFile* is the only program included with *Windows* which comes with two icons. Use the FILE / PROPERTIES / CHANGE ICON command within *Program Manager* to choose the icon.

WINVER.EXE—This program shows the version of *Windows* and the mode it is running in.

To install the icons for the four programs listed above, type: {ALT+F}, {ENTER}, {ENTER}, {TAB}, then the path and filename, e.g.:
C:\WINDOWS\MSDOS.EXE
MSDOS.EXE and WINVER.EXE are in your *Windows* subdirectory, and the other two programs are in the SYSTEM subdirectory.

Tip #2: Compact *Windows* for use on a laptop or a crowded hard disk.
The following files may be deleted safely if you are short of disk space:

**.BMP*—Bitmapped graphic files for use as *Windows* background. They are quite large!

**.TXT*—After you have read them, all the countless README files that come with *Windows* can be zapped, too. Be careful not to delete your Notepad files which end in .TXT.

**.HLP*—The help files, while useful, are not essential. You can selectively remove some of them—e.g. for the games or for *Notepad*.

TERMINAL.EXE—If you have a laptop without a modem, *Terminal* is a sheer waste of space.

PBRUSH.EXE—practically useless on a laptop without a mouse.

KRNL386.EXE, WIN386.EXE, SWAPFILE.EXE—you don't need them on a 80286 or 8086 machine, and they take up more than 500 kB.

WINFILE.EXE—You might want to use MSDOS.EXE instead and save 100 kB. About the only important thing MSDOS.EXE cannot do is format floppies. The old one (ver 2.xx) could. Progress...

If you have an AT laptop and dump the files listed above, plus a few of the *Windows* applications that are not essential to your work, such as CLOCK.EXE or REVERSI.EXE, *Windows* will occupy less than half of the original space!

Tip #3: Place *Windows* on the first cylinders of your hard disk to reduce seek times dramatically whenever *Windows* accesses the drive, even if you use a cache. To do this, install *Windows* on an empty hard drive, delete unwanted *Windows* files (see tip #2), run a disk optimizing utility (*VOPT*, etc.), then install all other programs.

Tip #4: Speed up the hard disk by shortening the PATH statement in AUTOEXEC.BAT, thereby reducing the time DOS and *Windows* take to locate a file on the disk. Most *Windows* applications add their subdirectory to the PATH statement during installation. You can remove this and replace them with the name of a subdirectory in which you keep simple .BAT files you can create to launch the applications. My PATH statement lists only one subdirectory, which houses a couple of programs and batch files (one of them starts *Windows*). *Windows* itself doesn't have to be in the PATH either!

Tip #5: (For laptops) Use *Windows* Recorder to create keyboard macros for mouse movements, rather than using a trackball or another poor substitute for a mouse. All menus in *Windows* have keyboard shortcuts, but many actions are performed faster with a mouse—e.g. moving and resizing windows. The macro recorder can memorize and repeat all your mouse movements—you can record the movements at home, then play them back on the road with a single press of a key. Attention! For reliable results keep the border width of windows no smaller than 3mm.

Tip #6: (For laptops) Change the colour of the *Windows* background to black and solve The Case of the Lost Cursor on an LCD screen. Unless a laptop has an active-matrix (TFT) technology display, the mouse pointer will disappear every time you move it. With a black background the pointer will be white, and if you lose it, just move the mouse down and a few millimeters up—the pointer will show up somewhere near the bottom of the screen.

rom the Trenches

by Steve Bellamy

Last month, we introduced the first of Steve Bellamy's columns with an incorrect attribution. His company is indeed Mugen Computer Consultants, but far from being 0473-78-4887, his telephone number is much more like 0473-78-4667. These days he's in training for the 1992 U.K. National Aerobic championships, so he might appear to breathing heavily when you get him on the line. Fear not, but instead...

Transcend and be productive

"To laugh at men of sense is the privilege of fools."

● Last month we took the first **4 Steps to Liberation** from classical computer misuse. Let's move on...

Please don't spend time, energy, or money making your computer better—rather, let your computer make *you* better. Computers are not word processors, spreadsheets, accounting machines, multi-media presentation creators, or any other such nonsense. They are simply computers. They add numbers very quickly—period. If you want to make the very best of the amazing tool on your desktop, let it perform the classic three steps of creativity, guided by the best thinking engine in existence, your mind. Let it:

Record, Recall and Recombine

Step 5: Record. Start thinking in terms of the great mass of incoming details merely as individual NOTES to be dealt with regardless of their format. A question, an idea, a bill to be paid, a meeting to attend—they are all notes differentiated only by when they occurred in your life. Start to record everything in the computer in the same way. No categories, no keywords, no fields, no nutt'n—only the date it occurred. Everything is a note, everything gets remembered. Several programs do this kind of thing very well: *Agenda, Grandview, Instant Recall,* etc.

Step 6: Recall. Retrieve each note and deal with it there and then by either throwing it away ("it's a bad idea"), transferring it ("this is not my job"), taking some action on it ("let's pay that bill"), or filing it for later reference.

Step 7: Recombine. Take those three great idea notes and write that compelling article, letter or proposal. Take those five good restaurant tips and make a good eating guide. When you've finished, the new combination note becomes another item for later recombination. And so on.

Step 8: Read my column next month. Now we're getting somewhere. Furthermore, the live version of this column will be available throughout June and July in the form of the **Tokyo PC Beginners Course.** Contact your local Tokyo PC office for further details.

● Meanwhile ..... **Ok, know-it-alls,** why is the screen 80 columns by 25 lines? (Put your hand down, Mike.) Answer next month.

● Do you really think computers are designed by careful-thinking pipe-smoking benevolent **Albert Einstein look-alikes**? Hell, no. We all know about the QWERTY keyboard being designed to slow fast typists down, but did you know that when floppy diskettes were first made on contract by IBM for the National Security Agency, some of the specifications were that they had to be disposable, and that it should be easy to destroy the data on them? IBM could have made them virtually foolproof, but the spooks worried about possible retrieval by hostiles and unfriendlies. Peter Norton has made a fortune from, and we have had to live with, that little fact....

● **Steve Drops Famous Names Dept.**—I just have to tell you this, privacy laws notwithstanding. There I am on Compuserve, having posted a rather simple question about communications. I wasn't too clear about the benefits of Ymodem or Zmodem or something like that: could someone please tell me what the difference was? Somebody did, and his name happened to be Ward Christensen. What a coincidence, I thought—maybe this guy should hear about his namesake, **the near legendary Ward Christensen**, who just about created the PC telecommunications business by writing XMODEM and setting up the first BBS in Chicago. So I wrote back a rather condescending little message to that effect. He wrote back within an hour. Er... um... (much shuffling of virtual fee) well, folks, how was I to know he was still alive and computing? So, Peter, next time you have a clever and witty comment on Xmodem, Ymodem, or the like, take it to the source *[No, no, not The Source—CompuServe. Ed.]* and ask 76703,302. I'm sure he'll be just as helpful to you as to me.

● One of the most valuable tools you can use in your computer is a **Macro Recorder.** Try to set it up so that whatever you do on a regular basis can be done by a single keystroke. Most major applications come with amacro recorder built in, but I prefer to have a separate utility running independently. Productivity takes a real boost when you start to automate such mundane tasks as Alt-S)aving, Alt-B)acking up and Alt-P)rinting hard copy. My tool of choice is *Keyworks*, but such others as *Prokey* or *Superkey* work just as well. The nice thing about *Keyworks* is that it allows you to set up menus to automate your macro usage, a real help. *["Macro automator?" Anything like a "lens magnifier"?. Ed.]* However, you should avoid Ashton Tate's *Control Room* macro facility. It has a life of its own—not the sort of thing you need in a macro system!

● **Last month's column** got me quite a few phone calls, even though my telephone number was printed wrong! Most of them were very positive, and I really enjoyed talking to you folks. I even got a couple of calls from "concerned individuals" working in local branches of large U.S. software companies, saying how things in Japan have changed recently, and how they had a new positive attitude focused on helping end-users. I called one the next day, but he was in a meeting with no end in sight, and his fluent English-speaking secretary told me he would be away for two weeks on holiday (I didn't ask if it was Kyoto), and so would I please speak to one of their technical staff? I did...see last month's column.

Dances with DOS

by Wm Auckerman

In less than a decade, the personal computer has progressed from novelty to ubiquity. As part of this process, for many of us the computer has evolved from a luxury to a necessity.

I have both a laptop and a desktop. At a conservative estimate, I spend 10 hours a day sitting in front of a computer screen. Much of that time, I'm working. When I'm not working, I'm calling a BBS or playing a computer game. Then there's the time I spend reorganizing files or experimenting with parameters and patches. And I make occasional forays into programming, exploring the innards of application files and RAM, or doing any of a dozen other computer-addict activities. And you know what? I love it (even the work part, sometimes).

I've been a computer user for more than seven years, and computing still holds an allure that the years have not dimmed. (Yet, I've never felt the "romantic" urge to "personify" my equipment. I know some people who have "named" their computer and affectionately refer to "her." I've resisted such anthropomorphic tendencies; my computer is, and always will be, an"it.")

Sometimes, I have trouble recalling what I did with my time in the years BC (Before Computers). I suppose the major tradeoff has been that I rarely watch television anymore. The computer screen holds more fascination for me than the TV screen.

With that as background, it won't surprise you to learn that I take exception to some of the comments made by Steve Bellamy in his column in the May newsletter. Whether my experiences serve as a rebuttal or a justification of his basic premise, however, may depend on your point of view.

Steve has suggested that "computing should be one of the most boring and mundane jobs you do." The central (though unanswered) question raised by this statement is: "why?" Why should I transform what is still an entertaining, transcendental experience into a boring, mundane task? My life already has enough boring jobs (washing dishes comes immediately to mind)—I see no reason deliberately to create another. I prefer that my computing time remain a delight.

"Why does everyone get so excited about computers?" Steve goes on to ask. For "computers" in the previous question, substitute "sports cars," "birdwatching," "stamp collecting," or any of the uncounted manias that infect subsets of the general population, and the answer becomes clear. In the words of Louis Armstrong, "Man, if you gotta ask you'll never know." (Without that sense of excitement about computers, after all, it's doubtful that TPC or this newsletter would exist.)

Steve may be correct when he suggests that his advice would help me to become "more productive." His prescription mistakenly assumes, however, that maximizing productivity is among my goals. Let the MBAs quantify productivity! I prefer to use qualitative yardsticks to measure my computing hours. It's not just how much I can do; it's how well I do it and how much I enjoy it.

I've resisted such anthropomorphic tendencies; my computer is, and always will be, an "it."

Enough polemic for this month; now it's time for an admission. In spite of my daily dances with DOS, I have not yet become proficient at dancing the hardware rock. I can hold my own in the software samba, and I can do an especially mean word-processor polka. But when the synthesizers strike up heavy metal tunes with lyrics of VGA, Hercules, SIPPs, and LANs, I reluctantly yield the floor and join the wallflowers.

Unfortunately, there is no Arthur Murray school of hardware, guaranteed to teach the basics of cards and peripherals in 20 lessons or your money back. Learning about hardware is a do-it-yourself proposition, and the logical place to start is with your own system: What CPU does your computer use? What is your BIOS release date and ID? What kind of adapter does your graphics card use? How many COM ports do you have?

I've seen several programs that can answer some or all of these questions. One of the better of these is *YOURSYS*. No longer need your IBM remain just a big blue box; *YOURSYS* will allow you to figuratively open the cover and peek inside.

YOURSYS generates a report about system hardware and software in 11 categories. As an example of the type of data it reveals, the listing in the box shows what *YOURSYS* version 1.2 has to say about my Dynabook (J-3100SS) for three of its categories. Other *YOURSYS* categories provide information about memory usage (total/free conventional, expanded, and extended memory), disk drives (type and total/free space), a simple memory map (program, size, interrupts), environment, CONFIG.SYS and AUTOEXEC.BAT, and (for Windows 3.0 users) SYSTEM.INI and WIN.INI.

A couple of minor points. The date stamp of the YOURSYS.EXE file is 3/91, but a quick scan with a list utility suggests that *YOURSYS* cannot identify a 80486 CPU. And though it can recognize a V30 CPU, running *YOURSYS* on a 9801 causes a system lockup.

YOURSYS isn't the kind of program you'll use more than once or twice, and the information it provides won't make you an instant hardware guru. But if someone should ask, "Does your system have an EGA card?" you'll be able to answer "No, it's VGA" instead of "Gosh, I don't know."

YOURSYS Ver. 1.2; available as YSYS12.ZIP on TPC & P&A BBSes; freeware ($0); Copyright by Gateway Software, Inc. 61 East 8th Street, Suite 128, New York, NY 10003

Of Mice, Murphy and Minnesota

by David Parry

There I was, preparing to head off to Minneapolis, and facing my very own 3M. The disaster-plagued April newsletter was ready for printing when my mouse suddenly upped and refused to acknowledge its existence. *Ventura Publisher* requires a mouse, so I tore my hair yet again and called the ghostly Andreas Braem, the eminence grise of DTP at TPC. Chris Case had already contacted Andreas and was Kamakura-bound in his Murphyesque Volksmobile, the one he had virtually lived in for the past two days while traveling in and around Tokyo. On my return two weeks later, I finally exorcised all incarnations of Murphy. My machine had been checked, and the mouse worked everywhere except *Ventura*.

In Minneapolis I saw the 3M building, and prepared to acquire the third M of my own saga. A flying visit through a software shop revealed a plethora of software, knowledgeable assistants and prices that made mail order a cheaper alternative even from Japan. The megastocks of the mail order boys give them real clout when ordering from Peter Norton or Lotus. I asked for *CorelDraw 2.0*, but it seemed it had not been released yet, except on probation to high-placed IBMers.

I headed off to Rochester, Minnesota, for my course at Metafile. The first week was spent in learning the relational database aspects of *Metaview*, and soon I was LOCATEing, RELATEing, and indexing with the best of them. The course proceeded rapidly, and could have been even faster were it not for the use of asthmatic PS/2 Model 30s with bronchial hard disks sporting access times of a mere minute or two. These clunkers should have checked in to the Mayo Clinic just down the road, where just about every other non-resident of sleepy Rochester appeared to be headed. "Second opinion heaven," said my friend in Minneapolis with reference to Rochester's main landmark; what indeed can one say of a town where every hotel has wheelchairs, many also equipped with holders for IV lines and drips?

Hold the Mayo.

That is not all there is to Rochester. Don't forget the IBM facility, which we visited on the final morning of our first 4.5 day course. IBM Rochester designs and manufactures the AS/400 minicomputer, successor to the very successful S/34, S/36 and S/38 series. We saw examples of each on display, plus hard disks ancient and modern. The production line was most unexciting, apart from the molding machines and an occasional NC lathe producing machine parts, but then this was not a car factory, so everything was clean and quiet. The metaphor could be extended to Rochester itself, which proudly flaunts its award as one of the top ten typical American towns.

Back on the tour, we saw clean rooms for certain of the processes. The standards could not have been all that rigorous, since the workers had face masks that did not cover their noses. However, a photo of the high-grade clean areas showed workers in astronaut-like uniforms with helmets and outside air supplies. At last they are learning from the Japanese. We finished the tour in about half an hour, marveling that people were already going to lunch before 11 a.m. It seems that production workers can clock in at any time between six and seven-thirty, within six-minute time bands to avoid a crush of traffic. The Germanic origin of the good citizens of the area was clearly at work here. Not since my student-era stay in Stuttgart had I seen such a strong *early to work and early to bed* mentality. Nor had I seen offices emptying at five-thirty for many a moon, and I suddenly yearned for a hefty shot of midnight oil, Tokyo style.

Oil there was no shortage of, for the well-fed citizenry enjoyed fine dining and plenty of it. In the evenings we cruised the hostelries of Rochester on our expense account, bemused by the stupendous portions. Mammoth steaks flopping over the plates, huge salads overflowing with goodies, and side dishes that would have been complete meals in Tokyo. My plaintive requests for a small dessert produced either a plentiful dish of fruit (for some reason, it was always the same—three kinds of melon) or a one-scoop ice cream. The scoop they used must have been borrowed from a drag-line excavator at a Montana coal mine, so my digestive system was taxed to the utmost until I got used to the fact that the Minnesotans were utterly determined that no diner should ever walk out hungry. I sought to avoid this excessive hospitality by searching the menu for "healthy dishes for the day before your medical examination", but even those would boost your lipids and cholesterol to red-lined levels.

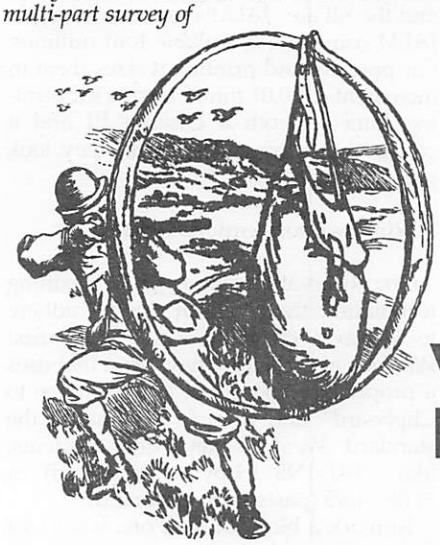
The first weekend came, a break in our DOS boot camp routine. I flew from the bright lights and bountiful restaurants of Rochester for the reformed sleaze of the Twin Cities, where municipal clearance schemes had removed much of the picturesque inner-city attributes that I had expected to see. No slums, no Uzi-toting drug dealers, no pimps in white Caddies. The refugee problems of the world were mirrored in the restaurants. Lebanese, Ethiopian, Vietnamese, many staffed by Hmong from Laos. Perhaps the Midwest aspires to be just as *akarui* as Japan, with relentlessly cheerful people in bright new buildings or reconstructed "historical" buildings barely a hundred years old.

The second course covered the document processing side of *Metaview*, with workshops on scanning, and image retrieval. *Metaview* allows the user to store and retrieve images from optical or hard disk, with associated indexed information entered in the usual alphanumeric way. The database section was *dBase*-like, but more powerful than *dBase*, according to the more experienced people within our group.

By the end of the week we were looking at LANs and mainframe links, topping off our course with a visit to an office that was using *Metaview* for fax handling and document scanning and retrieval. In a nutshell, fax just becomes another image, in the same way as scanned input. A Gammalink card provided the PC fax capability of the system, although *Metaview* itself is not tied to any particular hardware.

I had arrived in Minnesota on a thundery but cold day. The IBM tour took place on a day that went from wintry to summery within hours. I then left for St. Paul, where the temperature soared to July levels within a day and then dropped again almost as quickly. When I turned up for the second course on Monday morning, winter was reasserting itself, and the rest of the week was spent in wet, cold and cloudy conditions, enlivened only by a snowstorm.

Last month Peter Evans took us on a tour of the cyberbin, and this month concludes his multi-part survey of



vercoming Depression

Apart from its surprising inability to take some of the simplest hints—such as a question mark at the end of a sentence—*Overcoming Depression* does a good job of parsing input and an uncannily good one of formulating grammatically acceptable answers. Yes, one does get the impression of a dialogue with a—with a *something* that has interests and, unlike *Racter*, sanity.

Three months back, I gamely simulated extremely mild depression—mostly by exaggerating aspects of my customary grumpiness—and plodded through introduction and all seven chapters of *Overcoming Depression*. All went well, and grumpiness started to lift, until I reached "medication." Here I subjected myself to a questionnaire about my psychic disorder and its physiological manifestations, which I answered truthfully. I was astonished to learn that I would probably benefit from being doped for a few months. Further experimentation led to the amazing conclusion that *any* set of answers brought the same recommendation.

Very surprised, I quickly left the program, on the way engaging in a delightful dialogue (for a sample, see "One Track").

When asked for comments on the program's enthusiasm for medication, Dr Colby seemed as incredulous as I had been, and sent me "the revised version."

It certainly had been revised: in one year the language data files had jumped from 583 kB to just over 1 MB. The program no longer recommended chemical additives. However, I was saddened by its total lack of interest in any comment I made on exit—it seems a new bug locks out the possibility of dialogue at this point. Perhaps the program is being revised a little too rapidly for its own good.

Where does all this leave *Overcoming Depression*? The underlying theory seems plausible and the version on screen clear and persuasive. The dialogue is enjoyable and reassuring. The advice on drugs—which is anyway accompanied by a number of prominent disclaimers and warnings—now appears sensible. But would it be convincing? Would it work?

I infer that the program is a descendant of others used under Dr Colby's rather more direct supervision. Most of his patients might then have been entirely innocent of computers, needing only the ability to type. I wonder what they would have thought of such "crucial concepts" as We Can Program And Reprogram Ourselves As Adults, overriding and overwriting the old programs of childhood, which contain bugs (errors).

One Track

OD: You enjoy the chance to say anything you want here.

PE: Yes, and you do too.

OD: We have covered a lot of ground here. I hope you will go over it many times to absorb the information.

PE: I shall. I shall look at the files. Will you do so as well?

OD: Your printouts now comprise a small textbook on your depression.

PE: Yes. Will you read them to me as I lie in bed?

OD: What interests you the most sexually?

Some care has gone into explaining said crucial concept, but I should guess that such a mechanistic model of personality, however valid, would be too distasteful to many people to be therapeutically helpful. To me, it is melodramatic—the first images it evoked were from *The Manchurian Candidate* (plucky Americans tortured by North Koreans, or United Artists). Moreover, it is an unfortunate reminder of the total predictability of the program. For its vocabulary and other resources are on display to any computer user equipped with a tool as simple as *List* (see "Expanses of a Virtual Shrink" in last month's issue). This display emphasizes the program's limits, and removes the mystery and the fun. The data files should come in some inscrutable compressed form (like the dictionary of *Namegram*, reviewed in the April issue); as it is, *Overcoming Depression* will likely appeal only to those who have a computer but know very little about it—an unnecessarily small slice of society.

"Why not leave their private sorrows to people? Is sorrow not, one asks, the only thing in the world people really possess?" So asks Timofey Pnin, a man with good reason to be wary of shrinks, their smiles and their psychometrics.

The stolid computer, however, its memory and output completely within your control, has a certain appeal as a psychiatrist. At the simplest level, there is no reason for embarrassment in divulging your horrific Hydean peculiarities. Moreover, if you regard the computer as a linear-thinking coprocessor for your own neural network, discourse with it becomes a particular kind of soliloquy. Your private sorrow remains your own.

This advantage of the computer could be used more fully. Cognitive therapy, which *Overcoming Depression* both purveys and embodies, appears to be very effective among "one on one" therapies for dysphoria. But, as Beeman points out, an etiology likelier for many patients may prescribe "family therapy", where the therapist works on the social interaction leading to and aggravating the dismal status quo. Undepressed layman that I am, I can only guess that the truly miserable person finds it easier to seek human help in indisputable privacy than to do so with the suspicion—or worse—that comments and speculations are being passed on to a spouse. In view of which, I envisage a "multiuser" (though single-tasking) version of the program, or at least a version for two people. This would give each person a password to his or her log file. Responses by each would feed a single encrypted data file, this in turn influencing the questions asked and comments given the other—but doing so indirectly and discreetly.

Who knows. The good efforts and inevitable gaffes of the bumbling program in dialogue mode might lead our hypothetical miserable pair to comment, talk, chat, even laugh—and to overcome depression.

Overcoming Depression
by Kenneth Mark Colby and Peter M. Colby
Requires 1.4 MB disk space and 530 kB free RAM, list price $199
Malibu Artifactual Intelligence Works
25307 Malibu Road, Malibu, CA 90265
Tel 213 456 7787

Martin Bruczkowski

Bikan

A.D. CDLXXVII. The remnants of the Roman army are hunted by hordes of wild Germans. Hot on their heels are the barbaric Huns. Absolutely no PCs are in use, not even Z-80s. At the same time in Japan, a new and wonderful invention creeps in from China through the Korean Peninsula—KANJI!

Who would want to limit one's writing system to an alphabet of a measly 26 symbols, when one can have 2 syllabaries plus a few thousand complex little nightmares? Maybe the ancient Japanese were farsighted enough to plan the booming sales of soft contact lenses for the tired eyes of the 20th century student of *kanji*, but in all their wisdom they did not predict the End of Japanese Domination over the World of Technology: 1981 and the Birth of the PC.

To give credit where credit's due, the Japanese did invent a couple of (totally incompatible) systems, relying on fonts in ROM (both PC and printer). The past few years have even seen the emergence of bilingual machines, e.g. Toshiba and AX models. But only very recently have *kanji* and *kana* found their way to our old, faithful IBM clones. One of the first in the field was the Japanese word processor *EW Plus*, infamous both for its price and its dependence on a little gizmo that sat waiting in a parallel port to start conflicts. The proud Americans, always wanting to prove that they can beat another nation at its own game, answered with *MOKE*—a shareware *kanji* editor which, although simple and cheap, certainly did the job of both displaying and printing *kanji* using a plain IBM or clone. But what about a product for those who want only an occasional Chinese character in their ASCII text or graphics file, and who want it to look good? Who would make such an extravagant demand?

I, for one, on a number of occasions. The first one was during my days at Sophia University, where, as the proud owner of an AT clone and a dot-matrix printer, I submitted all my papers on the gems of Japanese literature as printouts, except for Japanese names of places, persons, and books, which I had to pencil in. Then came the professional days, when I taught Polish. The LaserJet produced beautiful posters advertising my lessons, but the Japanese part—pencil again.

The times they are a-changing. The product has arrived to answer the needs not only of Polish teachers in Tokyo—Knox Computer Systems' ***BIKAN JALM*** or ***Japanese Arts & Letters Manager***. *JALM*, being a standard *Windows* application, can display and print Japanese characters on its own, using any *Windows*-supported printer. The catch is: little or no control over the look of the printed page. Thus a graphics package, like *Arts & Letters*, *CorelDraw*, *Adobe Illustrator* or *Micrografx Designer* does help, especially if your goal is an advertisement or brochure design incorporating Japanese ideograms as well as other graphic elements and English fonts. You do not have to use *Arts & Letters*—all popular drawing packages for *Windows* do the job as well, which Knox honestly admits in the manual.

The *JALM* files occupy roughly 3 MB of space, most of it by *kanji* font outlines—you will receive perhaps another 2-3 MB of them after you send the registration card to Knox.

Any *Windows*-capable machine will do, although running *JALM* on our BBS, which is a 12 MHz AT, left me wishing I had a new Tom Clancy novel to read while waiting for the "DRAW ACTUAL" command to finish its page preview. As installation dawdles, you will be tempted to peek inside the manual. A nice surprise, dear *Windows* veterans; it's actually quite short, practical, and understandable, even if it smacks of translationese here and there. The table of contents lists every topic and command, so the lack of an index is only a minor inconvenience. My review copy came with two manuals, one written in Japanese and the other English. Installation finished and a new icon added to Program Manager, *JALM* is ready to run.

The software is really easy to use. Any experienced *Windows* user who has also used a Japanese word processor will master text input and editing in no time. The *hiragana* prompt in the *kanji* input area means the program is ready for input. {F2} switches to *katakana*, *romaji* or JIS input modes. The last lets you directly input characters from a JIS character chart (included in the manual) by typing their numbers—useful if you want to input certain symbols. JIS codes include most European-language characters, Roman numerals, and special symbols, as well as kana and all everyday-use *kanji*. Unfortunately, entering many of the JIS codes produces the message: "No font data for xxxx". I hope the missing fonts will be on the extra font disks.

Input of Japanese text in *hiragana/kanji* mode follows the rules of most Japanese word processors. It is visible onscreen—but only in draft mode. For WYSIWYG preview, choose the Draw Actual command (why, oh why is there no keyboard shortcut for that?), which reveals the shape, size and colour of the characters. The fine-tuning of the fonts' vertical and horizontal size, spacing, colour of outline and the fill are *JALM*'s greatest strength. *JALM* comes with scalable font outlines. For preview and printing it sizes them in increments of 0.01 mm. I have tried printing *kanji* on both a LaserJet III and a colour dot-matrix printer, and they look great in any size

Windows conventions flouted

One observation about *JALM*'s editing techniques: the program doesn't adhere to *Windows* conventions. It is the first *Windows* application I have seen that uses a proprietary command "Copy Image to Clipboard" and doesn't respond to the standard *Windows* keyboard shortcuts, like CTRL-INS (copy to clipboard) or SHIFT-INS (paste from clipboard).

To mark a block of text, one has to set the start of a block at one point and the end of it at another. Most *Windows* users have a mouse or some other rodent, and clicking and dragging is not exactly a foreign technique. The next usual step after creating a chunk of text would be to save it. Welcome again to my pet peeve: where are the *Windows* conventions? Instead of "documents" or what "files" you have to deal with, you are offered *libraries* and *symbols*. A *library* can store up to 95 symbols, while a *symbol* can contain a single character, a sentence, or a large document. Why such complications?

I hasten to present a nicer side of our guinea pig—its Import and Export capabilities. *JALM* comes with filters allowing you to export its characters to the following formats:

Encapsulated PostScript (widely supported scalable graphic format)

EPS Tag Image File Format (a universal, bitmappedgraphic format)

TIF Windows Metafile (shared by several *Windows* applications)

WMF Shift-JIS text file (to be used with Japanese word processors)

SFJ

It really wouldn't hurt Knox to add export filters for .CDR, .PIC, .GEM, .WPG, .PCX and some other graphic formats. Still, the ones you get are usually enough—if your program works in *Windows*, it can accept letters from *JALM* through the Clipboard or the .WMF format. For other applications, TIFF or EPS usually does the job. If you are the proud owner of *Arts & Letters* you are better off

still, for *JALM*'s native format can be directly opened by *A&L*. *JALM* will also import *kanji* using the Shift-JIS, *zenkaku-moji* (two bytes per character) file format supported by popular Japanese word processors like *EW Plus* and *Ichitaro*.

What will happen now that DOS/V is out, allowing IBMs and clones to run Japanese software, and vice versa? This revolution in OS may pose a threat to *BIKAN JALM*. Indeed, it may force Japanese word processing programs upscale to compete with *BIKAN JALM*.

Jalm

Wolfgang Bechstein

A KANJI BUTTONHOLE?

As an infrequent *Windows* user but habitual *kanji* contortionist (= translator), I was asked to provide a few comments on *BIKAN JALM*. What does this program offer to the person interested in producing or handling *kanji* files on a machine without any Yamato ingredients? Well, not very much, I am afraid. But I am jumping to conclusions. Let's be fair and start at the beginning.

As Martin points out, the program is fairly easy to get up and running, and before long, the user will try his or her hand at kanji input. The kanji conversion routine which is an integral part of the program is adequate and about on a par with E1.SYS, the front-end processor of *EW Plus*. The processors that come with IBM's DOS/V as well as *ATOK*, *EGBridge* or *VJE-Beta* offer a few more frills and somewhat larger dictionaries (about 80,000 words vs. 55,000 words for *BIKAN JALM*), but a larger dictionary does not necessarily mean better conversion for everyday sentences. In *BIKAN JALM*, changing the phrase divisions for kanji conversion is simple, and a two-colored phrase cursor makes it easy to see which part of a longer string will be converted when one hits the space bar. Homonym selection also follows standard procedures, and a *gakushû* (learning) feature is provided. Of course, the program does not require a keyboard with a *kanji henkan* key. It works with the regular North American IBM set of ivories, since it uses the space bar for *henkan* and the function keys to switch between input modes. Unfortunately, *BIKAN JALM* does not support *hankaku* (single-width character) mode. All characters including Roman letters are *zenkaku* (double-width) only. It is also not possible to convert, for example, a string into katakana after the fact. The katakana mode has to be activated beforehand.

A bigger problem is the on-screen look of the kanji in draft mode (the only mode in which one can do anything). The program uses the "Small-Gothic" font which I find positively ugly and much harder to read than, for example, the "character-based" kanji text mode of DOS/V. As Martin describes, selecting the "Draw Actual" command, after considerable churning, does produce beautiful kanji in any size or color, but the on-screen view then is strictly an uneditable still life where all other commands, including of course kanji input, are inactive.

What I find to be the biggest single limitation which rules out the use of *BIKAN JALM* even as a rudimentary word processor is barely hinted at in the manual: "The document area always uses the Small-Gothic type font. The document area size is fixed at 14 lines and 39 columns. The document area cannot use the vertical/horizontal scroll."

It is hard to believe, but true. At first, I thought I had gotten something wrong. This is a *Windows* application, after all. Normally, windows open up onto a larger world, and even *Windows* windows commonly let you see more of a document, using the scroll bars or the PGUP and PGDOWN keys. Well, with *BIKAN JALM*, the text you see in that little 14 line by 39 character window is *all you get*! If the window is full, and one inserts text somewhere, the somewhat cryptic question "Clear bottom line data?" appears. When one has the mouse say yes, everything beyond the 14th line falls off the boat. Irretrievably! Worse still, although *BIKAN JALM* allows import of Shift-JIS *kanji* files, it truncates imported files without any warning if they are longer than 546 characters. Only the first scrap of the file will appear, and if one tries to scroll, well, see above. If one exports the file, it will consist only of the part handled by *BIKAN JALM*, i.e. 546 *kanji* at the most. As a small saving grace, the program does append a different file extension when exporting, so overwriting of the original file is not likely. But still, even a vintage 1985 *waapuro* can handle longer files (albeit not on an IBM).

When printing, *BIKAN JALM* offers no indents, page numbering (unnecessary in view of the document length) or any other page control features. Therefore, insinuations in the manual to the contrary ("*JALM* is a Japanese word processing and publishing tool"), this certainly is not a word processor. At least not for very many words. What *BIKAN JALM* is good for is entering a few lonely *kanji* and either printing these (beautifully, but *slowly*) on a LaserJet or other non-*kanji* printer, or exporting them as graphics files for use with other graphics programs. Exporting *kanji* as text into other text-oriented programs, although possible, does not make much sense either, since such programs either will not be able to display or print them, or if they are and therefore operate in a Japanese environment, their *kanji* input and handling capabilities will be much better to start with.

Also, I found printing astonishingly slow. On my 25-MHZ 386 and LaserJet III, *BIKAN JALM* presented the "Printing" message for several minutes before it even sent any data to the printer. In all, it took about 5 minutes to print out a few lines of *kanji*. While this message is displayed, one cannot even switch to other *Windows* applications (so much for multitasking). As a consolation, the output *does* look very good, and *kanji* can be scaled without loss of quality. *EW Plus* also prints *kanji* on a LaserJet (and it seems positively fast by comparison), but *BIKAN JALM* is far superior in looks.

But are looks all that counts? For designers of advertising brochures maybe, but not in the dull word-crunching world where I'm making my living.

BIKAN JALM
Knox Computer Systems, Inc.
10055 Barnes Canyon Road, #K,
San Diego, CA 92121
Tel: 1-619-535-0771 Fax: 1-619-535-0773

SECRETS ENTRUSTED
TO A FEW

The Unpublished Facts of Life BEFORE TPC

as revealed by WICK SMITH Co-founder emeritus

TPC: How did you happen to get into computers in general, and computing in Japan in particular?

Wick Smith: After I graduated from [Antioch] college in '71, which taught me to do absolutely nothing, I spent a little time as manager of a theatre company. I had a so-called minor in communication, and I sort of hung around the radio station for most of the time. I was very heavily into audio production and script writing. Though I had a crappy announcer's voice, I did a lot of announcing. Then, broke and convinced that I was good for nothing in the world, I walked into Radio Shack when I saw a Help Wanted sign before Christmas. I found I was pretty good at selling people gadgets and gizmos. I was pretty interested in gizmos, although Radio Shack made some of the worst. I sold people the Cube clock radios, and those fiber optic light kits. I could tell any truck driver why an air suspension speaker gave you better bass, and why you have to have 80 watts per channel because 30 watts isn't gonna drive that, you know. I'd talk to them about low efficiency and high efficiency, and the guys are nodding very wisely. Very big in car stereo, but what really made me rich was that by the time I started getting into the heavy commission schedules, the CB thing hit, and Radio Shack was very well placed to do that—and the police scanner thing. We sold police scanners and CBs just the way MacDonald's sells hamburgers—over 3 billion sold. We kept changing the numbers every day; supply was our only constraint. So between hi-fi, police scanners, and fiber-optic kits, home alarm systems, things like that, it was pretty good business. I was running around in my BMW feeling like I was a real professional. Then I began meeting some of the guys in the Radio Shack organization, a little further up the chain, and these guys were on a whole different wavelength. What they were really into was making big bucks, but they were awful people. Their whole thing was to spend 18 hours a day at the store, don't worry about your personal life, 'cause this is your chance to get ahead, and basically everyone there was...nobody there had even a high school education—they just were excellent salesmen. Their only value in life was moving boxes, and they didn't care too much what was in the boxes. And I was, actually, kind of enthusiastic about selling somebody a fiber-optic kit, or making sure he got the good air suspension speaker, so I began to feel Radio Shack wasn't the place for me.

So, I went and got a really poor paying job at a high-end audio store a couple of miles away. There we were selling the Klipschhorns and Marantzes...

TPC: Where was this?

WS: Mount Kisco, N.Y. And of course, I left Radio Shack just when the first TRS-80s were appearing. You could actually buy a little computer kit from Radio Shack.

TPC: Didn't the Tandy initially come through them?

WS: That *is* the Tandy. But in those days Tandy was just the owner of the company. Tandy was actually a leather goods company that bought the thirteen foundering Radio Shack stores that were about to go belly up. Tandy just figured that Radio Shack was an ineptly run organization. They had a catalogue like Allied or Lafayette—the two catalogue monsters. But Radio Shack was famous for one thing—that is, being out of stock of anything you wanted. You went into a Radio Shack, everything they had you didn't want; anything you wanted they didn't have.

Tandy had made a fortune. I think he started with one leather goods shop in Texas. They made those leather kits—you know, where you buy the leather and cut it out and stick the beads on it—

TPC: Make your own keychain..Post-gimp.

WS: Yeah, that kind of stuff. Keychains, little wallets you sewed.

WS: He made a small fortune doing that thing. His big consumer insight was that people want something to be in stock when they go to a store. What he realized was that Radio Shack had the traffic—it

was just that they were continually disappointing people. So he bought Radio Shack and instituted his own very stringent inventory control system and supply system, developed a lot of sources in what was in those days called "The Orient," meaning Japan essentially, and got Radio Shack going. Every year there were another 100, another 200 stores—when I left there were a couple of thousand stores. Probably many more today. The real heartbreak for a manager was that you could have this enormously profitable store, and then Tandy would open four or five other stores within two miles of you. Your traffic would be wiped out overnight. So when you were made a Tandy manager, your days of being rich were numbered. I began to think that I would be good in the communications

I was a good writer, but I had this massive inferiority complex about people who worked in the city at "real" jobs and sat at desks and thought about stuff.

area. I couldn't figure out a way to get into advertising; it seemed like a bunch of very slick, smooth guys who all knew the right stuff to say and were very clever and so on. In fact, I was a good writer, but I had this massive inferiority complex about people who worked in the city at "real" jobs and sat at desks and thought about stuff.

TPC: The Madison Avenue crowd?

WS: Yeah. I mean, I loved the idea of making your living with your own wit—I thought that was a great opportunity. Anyway, I made a half-hearted attempt to get into that. Saw an ad in the New York Times for a copywriter for Tokyo. So...sounds good. It said "Send samples of your work." I'd never worked for an agency, didn't have any samples. So that night I wrote three ads. One was for Senheisser headphones—I can still remember pretty much the whole copy platform—one was for a telephoto lens, and the other one was for...I think it might have been for some kind of tape recorder.

TPC: The sort of things that came from Japan?

WS: Well, I chose things that I knew about. I knew about headphones, and I knew about cameras and things like that. I had no idea what this company did, and for all I knew they could have been the agency for Noritake china. Well, it turned out that this little advertising agency had the "Nigh-kon" camera account, or Nikon, as we came to know it. They had Pioneer Electronics, and they had a little company that I'd never heard of called "NEC." Apparently they got some much more professional samples of work from other guys, but what they saw in my work was...

TPC: You knew the stuff.

WS: That I knew the stuff. It turned out they were less interested in quality than in not having to reinvent the wheel. So the guys came to New York, interviewed me and hired me, and the next thing I knew I was in Japan—so that's how I got started there. Through the contact with NEC I started getting interested in computer things. I went there in '77; just about '79 some the CP/M machines were starting to appear. But I got started because there was a thing called the IBM memory typewriter down the hall... I began doing my copy on it. There was a little dial on it, and what you could do was to set it on Record and type something in, and after about a page you could save it to Position #1, and then you could turn to Position #2. So all the other copywriters, when there were revisions—and in Japan there are *always* countless revisions—had to type everything all over again. I would go to the memory typewriter, dial up Position #1—there was no screen on this thing. So you had to press the printout key, print to a certain point, stop it, and type in the part you wanted to revise, then you'd pick up later, and it would—it was very clumsy, but you could somehow store the stuff that didn't need revising, and just redo the stuff that did. So I began spending a lot less time doing revisions than my colleagues.

TPC: Early word processing.

WS: Very early pre-word processing. And then I discovered this thing called the IBM Displaywriter. Which blew me away. I was a converted IBM Displaywriter freak. You know, it had a nice screen, and things would reformat, and, you know...more and more I got hooked on the whole idea of computers as word-processors, the kind of work it would save.

So when the first MS-DOS machines came out, I determined I would have to have one, so I came down to Hong Kong and met this magical Englishman who sold me this computer called the Victor 9000 which had this incredible thing that I'd never heard of called a "hard disk".

TPC: A Winchester!

WS: Winchester, exactly. I said, "Hey, that's for me." So I got it. When I first met the guys in the PC club, I was the first and only guy that had a ten megabyte hard disk actually *in* his computer.

TPC: Who were the people who started the TPC?

WS: There was Rick Sheldon, who was an Apple II guy—one of the pre-Mac guys—who was basically seduced over to the PC side. He was working for the United Arab Shipping Co. or something like that. He and Steve Campbell and this other guy, whose name was Gordon Nebeker, who I think was the first president.

Gordon and Steve Campbell were the two guiding figures of it. And they really represented what I would say exists even today as two different strains of users. Gordon was very much the pragmatist, interested in getting the job done, constantly befuddled by hardware problems. He never couldget his disk drive to quite work right; something was always wrong with his display...Gordon could never figure out why he would ever need more than 128k of memory, etc. But he was very interested in this program called *Volkswriter.* And he was, I think, an early spreadsheet user, very interested in discovering the various ways software could help him do stuff. Steve, on the other hand, right from the very start was fascinated by the hardware. He was a guy who was constantly buying third party hard disks and going through the painful process of matching them to a disk controller, getting DOS to recognize that much hard disk. DOS had a hard time recognizing any hard disk, back in The Days of DOS 1.0, when there were not even any directories. You simply had the letter C:, and everything was in C:.

When IBM introduced the XT, at the same time they introduced DOS version 2.0, which had this bewildering new wrinkle called "directories". None of us

Steve actually would remove the hard disk, place it into his refrigerator the night before, or his freezer, and then bring it out in the morning.

knew why we would want "directories". Of course most people who saw DOS 2.0 didn't have a hard disk, so the idea of directories was just foolish. You had a 360k floppy. What on earth would you want to have subdirectories on that for? We were told it was a way of organizing files, something—didn't make any sense to us.

So anyway, Steve was into using hard disks quite early; and these were big outboard monsters connected to the machine through cables into controllers whose makers have long since died away, it seems. Steve was fascinated with all the complexities of entering binary code into the controller to get it to recognize the number of tracks, the number of sectors, heads, platters, and all this stuff. The whole notion of there being a physical drive with logical locations and stuff was something Steve was into pretty early. The rest of us were fascinated, but we didn't have the hours and hours of midnight madness that Steve put into this.

TPC: Nor, perhaps, the freezer capacity in your refrigerators.

WS: Right. Well, Steve had this one, very troublesome, hard disk, and he had done everything right, and he realized that in the morning, when he first woke up and turned on the machine, that the drive would seem to perform OK. He would test it quite rigorously; he felt quite good about it. He would come back after breakfast—Son of a gun, there'd be "Sector Not Found," "Error reading Drive C:," "Cannot write to Track." Oh-oh. Could not figure this out. He tried everything, and gradually he began to realize there was a direct correlation between the temperature and the disk performance. So Steve actually would remove the hard disk, place it into his refrigerator the night before, or his freezer, and then bring it out in the morning, and he got an extra few hours of performance out of the thing [*breaks up laughing*] before it warmed up.

Steve and Gordon met in Gordon's house before there ever was a Club—two bankers totally confused by this technology. They were both fascinated with it, and formed the nucleus of the Club when they began to realize that there were probably other lost souls out there who were fooling with this stuff. They posted some notices, or somehow got in touch with people, and we met a couple of nights, the entire club—about 6 people—at Gordon's house. Then we eventually found the Tokyo Union Church, which at the time seemed impossibly huge. That basement—it just seemed ludicrous for us to have all nine of us meeting in this—

TPC: Wembley Stadium?

WS: —Wembley Stadium of a place. But I remember one night a guy who worked for IBM, and who's long since gone back to the States, who was an avid Microsoft Word fanatic, showed us the first version of Microsoft *Word* we'd ever seen, which was really the first word processor that had anything of a graphical representation of a page. Somehow we were able to use one of those large screen overhead projection systems, and we had publicized it pretty well, and I remember being just absolutely astounded that night to find that about 65 people showed up. We had people standing; we had to go into the closets and drag out all the chairs they had. We had a tremendous show that evening. It was exciting, because first of all, it was technology we'd never seen before. The whole presentation method was something we'd never done before, the sheer number of people who happened to turn up to see it was enormous. I remember that night this tremendous sense of elation, like maybe this thing would actually go.

After that, we had some pretty sizeable meetings, and soon after that we actually began to have vendors wanting to come and talk to us. Some of those vendors gave some of the most atrocious evenings. Guys who could not put five words of English together, standing up, pointing to their instruction manual, constituted "our presentation for this evening".

TPC: That tradition hasn't entirely died out.

WS: I can imagine. We had some good presentations from a guy who was a distributor for Ashton-Tate, a guy who was quite instrumental in doing the very first conversions of software from English into Japanese. I think he did the original *dBase II* or *dBase III* Japanese implementation. Also, some guy attached to Lotus Development came over and gave, I thought, a very cogent presentation of what Lotus was doing in Japan. They were committed to the market, which at that time seemed a pretty bold step, in the sense that the Japanese had never embraced spreadsheets or the idea of executives or professionals using computers. In those days computers were very much a kind of a clerical entry tool for the Japanese—

Oriented around a computer, it was a little bit like having a refrigerator support group. You know, we'd sort of talk about compressors and door latches—"Does your butter tray have room for only six bars, or four?"

TPC: Real Men Don't Type.

WS: Exactly. A lot of OL's, heads down like a lot of monks, copying someone else's handwritten stuff and getting it up on the screen.

TPC: The men are supposed be doing the tough stuff...

WS: That's right. Sitting by the window and thinking deep thoughts.

When Lotus came in and told us they thought there was a market for professionals to be using spreadsheets in Japanese, I think there were some of us, and I probably include myself, who felt that, you know, maybe, but...it seemed a pretty chancey move. I don't really know how it's turned out for them, but I think they've done reasonably well; 1-2-3's done pretty well.

We also met with a guy called Ken Ishii in those days, who was from a company called ASCII, which was the company that formed an early alliance with Microsoft, became the Microsoft distributor, and then had a spectacular blowup, and the two of them went their separate ways. Ken Ishii was the guy who actually created the first laptop computers, the old Tandy 100, and the NEC 8023, or whatever it was called, and essentially put a lot of stuff into ROM and believed that there was going to be this movement towards miniaturization of computers. It was a vision, I think, that Gates shared, but the two of them had very different agendas. Gates later decided he was going to do it his way in Japan, and did. It was a pretty messy breakup.

TPC: Did you have a lot of SIGs early on?

WS: There was a time a couple of years after the TPC Club was founded when those of us who were concerned with keeping the club growing and moving and healthy were wondering whether or not the whole concept of a users' group wasn't in fact a little ludicrous—since a computer really is just an empty vessel. Oriented around a computer, it was a little bit like having a refrigerator support group. You know, we'd sort of talk about compressors and door latches—"Does your butter tray have room for only six bars, or four?"

I think essentially people are interested in what goes into and what comes out of a refrigerator, and I think it's the way they feel about computers, too. So what we tried to do was to institute a series of sub-groups based more around the actual contents of a computer—what people could *do* with it. There were those of us who were interested in the possibilities of using it with words. Other people were more interested in the databases. Don't forget that in those days our applications were fairly crude. We were limited to those Golden Three—Lotus, *dBase*, and *Wordstar*. We were all making the transition from *Wordstar* to *WordPerfect*. There

were a few early *XyWrite* buffs—I'm sure there still are.

We'd have these general meetings, and some guy would stand up to talk about—well, you know, in those days there wasn't any *FoxPro*, but there was something called *Clipper*, a *dBase* compiler, and three quarters of the audience would nod off, because to be interested in *Clipper*, you've got to care about databases and speed and a lot of issues that those of us who were totally into data communications...In my case, writing enormously intricate batch files, communications scripts: that's the stuff I was passionately involved in—how to automate all this stuff, so that you could create a message, push a couple of keys, have it automatically dial up, send out to the person you wanted, and log off.

TPC: We're talking about the early 80's, right—'81, '82?

WS: No, this would have been by this time, I guess, around '83, when we were getting into data communications—'83, '84.

TPC: Before the BBS began?

WS: Yes. Those were kind of exciting days. In Japan it seemed that everybody was either out of the country or about to leave or just arriving, so it was difficult to keep the schedules going, but as we grew to a fairly respectable size—75 or 80 people—we would break off into subgroups. We had 4 or 5 SIGs. There was a word-processing sub-group, maybe 20 or 30 people, that would meet in a smaller place. That seems like a very trivial and general subject today, but in those days it was something not many people knew about. The idea of actually, for example, using proportional fonts without having them run off the page—you know, it was a big challenge. A big Data Communications sub-group grew out of anyone's exposure to Joey Ito and some of the work of Jeff Shepard in the very early days of TWICS, and of course Steve Bellamy's work on Kanto Central. I recall in those days the Japanese were very, very reluctant to have anything at all connected to their telephone system that hadn't been approved and the performance severely crippled by bureaucracy. Hunting around Akihabara for a 1200 baud modem was just an exercise in futility. Acoustic couplers were the order of the day, because they weren't really connected to the telephone system. So most of us became telephone pirates; we learned what the red wire goes to and what the green wire goes to. We were using the 300 baud Epsom acoustic coupler with the famous Joey Ito switch to change from Bell 1.03 to CCITT. I think I've still got one of those in my closet, with the patented Ito switch on it. There was a kind of vitality about the club those days. I think some of the things that held the club together, and I'm sure it's so even today, were the Newsletter and the Bulletin Board. The Newsletter came first, because in those days we didn't know how to run a bulletin board. Steve Bellamy knew, but we didn't. And also the thing was that we were very reluctant to actually have the club own anything.

TPC: Ah, how to handle the finances?

WS: How to handle the finances, and in those days owning a computer of some reasonable power was a thing very much to be desired, so the idea that the club would actually own something, and an individual would have the use of this computer for the 29.9 days that the club didn't meet, seemed a kind of contentious issue. Little did we realize it would be more of a burden than a privilege to run that bulletin board.

I remember when Steve Bellamy had just opened Kanto Central in the San no Hotel. He had a little store inside the Sanno—I forget the name of the store—but he and I think somebody else got together, they ran the store, and he ran the bulletin board out of it. You had to kind of sneak into the store, because it was really officially a PX. It was a military installation, so you had to cut your hair and flash something that looked like an official pass to get into the place. And there in the back you could see this little AT, which was a very hot machine in those days, chugging away, and that was Kanto Central.

Little did we realize it would be more of a burden than a privilege to run that bulletin board.

TPC: Then it later moved to Steve's house?

WS: Yes, it moved to Chiba, which was still in the Tokyo area code. Maynard Hogg was our first bulletin board sysop. I don't think Maynard knew much of anything about data communications when he took it on.

TPC: How did you come to choose WWIV? Or was it the only one?

WS: No, we first tried RBBS. And there was one that was written by the Capital users group in Washington, D.C. I think that was RBBS—I'm not sure. We tried several. We tried one similar to the one that Kanto Central used, but what we found about most bulletin boards was that there seemed to be endless depths of menus to get to anything worth doing. Also there was a reliability issue. The other issue, which I'm not sure that you've solved, was for us whether there was any way to do multi-line. In those days the bulletin boards didn't really successfully support more than one incoming call at a time, so if the line was tied up, you simply had to wait.

But in those days we were charging an extraordinary amount for the dubious privilege of belonging to our group. We had a ¥10,000 initiation fee and the feeling there was that anybody rich and frivolous enough to own a computer would be rich and frivolous enough to part with ¥10,000. But we realized that there were students, 17 and 18 years old, who were beginning to take an interest in our club, and ¥10,000 just as an initiation fee, not including dues, was way beyond their means, so we made all kinds of provisions and except-fors.

The other thing that happened, too, was that there were people in Tokyo who knew damned well that they weren't going to be in Tokyo for 3 months longer, but they had to pay ¥10,000 plus dues just to attend 3 meetings. You could attend a certain number of meetings for free, but then you had to become a member, and we had sort of armed guards at the door to discover whether people actually "belonged". Then there was the issue of would we let anyone on the bulletin board, or only members; would members have more rights and privileges than non-members on the bulletin board? It seemed to me like the Executive Committee spent a lot of time worrying about stuff like that.

TPC: I don't think it's entirely finished doing so.

The Corporate View

by Roger Boisvert

Keyboards Move Indoors

When I first arrived in Japan, it was common in major office areas to see street vendors of typing services, just like people who hawk food and other services on the street. Japanese-language typewriters used to have about 2,000 keys, and only a highly skilled person could operate one of these typewriters at a reasonable speed. When typed documents were required, companies would send someone out to look on the street. It has been years since I have seen one of these street scribes. The adaptation of the computer to Japanese-language requirements put them out of business. Instruments with 2,000 keys have given way to computers and word processors of a mere 100 keys. Word processors may be seen everywhere in Japan now.

Keyboard Phobia Wanes

However, adaptation to keyboards took quite a while. The difficulty and amount of time required to type a document are still confining official documents and important documents to the handwritten. There is something personal about handwriting. Just as in the U.S., a graduation diploma from a university is usually handwritten, a handwritten document in Japanese carries associations of personal attention and value. Here, handwritten documents are still accepted where only typed or printed, computer-based documents would be accepted outside Japan. Because typing was so time consuming and provided so little added value, managers wouldn't be seen touching a keyboard. Anyone hoping to become a manager could not afford to be seen typing, and this included almost all male office workers in Japan. The task was always relegated to the lower ranks, almost always female clerks. Keyboard phobia and poor image is only now being overcome by younger men as Japanese-language keyboards and Japanese-language software have become easier to use, and word processors become more common. Yet, keyboard phobia remains prevalent among middle and senior managers.

Saving Face with Fax

Partially for these reasons, fax machines have been a standard piece of office equipment for years here. In fact, a person without a fax number would not be taken seriously in business as long as 5 years ago. The fax machine complements the Japanese business environment so well that it is hard to imagine doing business without one. A simple example. You are invited to visit a showroom. Japanese streets usually have no name, and there are no street numbers. You don't know where the showroom is located so you ask them to fax you a map showing how to find it. This takes only a few seconds, rather than several minutes' worth of dictation, "Take the Chiyoda subway line, get off at Hibiya station. Take exit A3. Walk to the first set of street lights, turn left and walk till you get to a small....." The usefulness of the fax machine reduced the need for extensive computer communications systems. Modems are still not in use at most companies, and some regulations regarding transmission of data are years out of date.

A common saying has it that it is easier to train a brand-new programmer in a modern language than to retrain an experienced programmer in an older language.

Computers and Waapros Converge

Several years ago, companies like Wang and IBM sold dedicated word processing machines worldwide. These dedicated machines lost ground to PCs as the latter developed the same capabilities and added others through software. In Japan, the trend was different, though equally logical. As PCs were introduced, companies producing *waapros* added PC-like capabilities to them. It is now the rare word-processor which does not have a built-in spreadsheet, address list, and time scheduler. Most of them also have communications capabilities. Elsewhere, dedicated word processors have all but died out, yet the segment is growing rapidly in Japan. I believe that as PCs acquire the same software capabilities, and as PC hardware becomes more standardized, the *waapro* will finally die out here. Today PCs and word processors are basically identical except for bundled software and a few specialized keys, which may make a dedicated word processor a more suitable choice than a PC for users with certain needs.

Manpower Shortfall Foreseen

MITI has forecast a shortfall of 970,000 programmers in Japan by the year 2000. While I don't dispute this number, I think it should be remembered that most of the programmers working at that time will be skilled in maintaining old systems written in old languages, and will be underemployed unless they are retrained. In the absence of retraining, the shortfall of programmers would be much more than the 970,000 forecast. Many efforts are being made to reduce the number of needed programmers by developing methodologies, languages, and systems with a much lower labour content. Results of the SIGMA project, development of CASE tools, prototyping, acceptance of higher-level languages, and use of 4D tools which deliver on their promises will be adopted by software houses and systems groups within corporations in Japan very quickly.

Good people like to put their skills to work, and in the future we will find the

best programmers working for software development companies, VARs and other software houses. This will put further strain on corporations that are developing products for internal use only. As software development companies gather the best programmers available and become adept at developing products for other companies, their ability to develop client-specific applications will increase. Once this happens, Japanese software developers should be able to begin producing world-class software for use outside Japan.

At the same time, the shortage of skilled programmers available for in-company development work will be exacerbated. Few companies are now capable of exporting application development skills. It is unreasonable to expect this situation to continue, given Japan's growing capabilities in application development. Japanese companies' ability to challenge foreign companies in their home markets will be partially dependent on their ability to hone client-specific applications development skills overseas. If these Japanese companies succeed, competitors like Borland, Lotus, and Computer Associates may have major problems on their hands. Several Japanese software development companies are reportedly eyeing the U.S. market. Though I believe most of them will not succeed in garnering considerable market share at first, the longer run is another matter. Game software aside, very few Japanese software packages deserve to capture U.S. market share. Do not expect this to continue. Japanese software developers will sooner or later be successful overseas. The one area where I see difficulty is in exporting the ability to develop client-specific applications using current tools. When software development tools become good enough to make customizing a simple task, Japanese VARs will begin to expand overseas.

Next month's CORPORATE VIEW will summarize points made in this series to date, then do a bit of crystal ball gazing to see what the future holds for PCs in Japan, and the possible effects on markets elsewhere.

40 YEARS OF CHANGE

An Historical Perspective of the Computer Industry 1950-1990

based on The Anderson Report

PLATFORM ENVIRONMENT

50's: Eniac & Sperry 1 (vacuum tube era)
60's: IBM 7090, 360/xx (rack mounted logic units)
70's: IBM 370, PDP-11, VAX (backplane with plug-in boards)
80's: IBM and clone PCs, Macintosh, engineering workstations (VME & Multibus)
90's: RISC (chip level multiprocessors)

STORAGE ENVIRONMENT

50's: Punch cards, tape
60's: Magnetic tape
70's: 8" and 5-1/4" floppy, Winchester disk
80's: 3-1/2" floppy, Winchester disk, cartridge tape
90's: Optical disk

OPERATING SYSTEM ENVIRONMENT

50's: none
60's: Proprietary (batch)
70's: Proprietary (interactive)
80's: UNIX, DOS
90's: UNIX, DOS-xx, OS/2, Macintosh

SOFTWARE ENVIRONMENT

50's: Assembly language
60's: Threaded code (COBOL, FORTRAN)
70's: Structured code (FORTRAN 77)
80's: Object-oriented (C & C++, LISP)
90's: High-level (knowledge-based)

APPLICATIONS ENVIRONMENT

50's: Banks, insurance companies, government
60's: Financial, government, manufacturing, engineering
70's: Engineering, small business systems, med. size companies
80's: Personal productivity, WP, spreadsheet, database, graphics
90's: Multi-media, graphics, visualization, communications

OUTPUT ENVIRONMENT

50's: Flexwriter
60's: Teletype
70's: Dot Matrix (Centronics), electrostatic
80's: Laser, color thermal
90's: Color laser

DATABASE ENVIRONMENT

50's: none
60's: CODQASYL (file/record structure)
70's: Relational (DB2, SQL)
80's: Relational (Oracle, Informix, dBase, Paradox)
90's: Object-oriented, knowledge-based

NETWORK ENVIRONMENT

50's: none
60's: RJE
70's: Time share (distributed data processing)
80's: Ethernet (distributed computing)
90's: FDDI (fiber distributed data interface)

VISUALIZATION ENVIRONMENT

50's: none
60's: Vector refresh (no color)
70's: Storage tube, Vector, low resolution raster (256 colors)
80's: High resolution raster (4096 colors)
90's: HDTV (16M colors)

SUPPLIER ENVIRONMENT

50's: Univac, Sperry, RCA, IBM
60's: GE, CDC, Honeywell, Burroughs, NCR, Digital Equip.
70's: Hewlett-PAckard, Prime, Cray, Tandy, Andahl
80's: Apollo, Wang, Sun, Apple, Atari, Silicon Graphics, Compaq
90's: NeXT? Sony? NEC? Nintendo? Toshiba? Fujitsu? Stardent?

oger eckons

With about 200 Toshiba laptops under my wing at my work, I have developed enough feel for these computers that when a sales rep has trouble determining what can or cannot be fixed, I often find I have a better understanding of the problem. But these are laptops, and recent experiences with a pair of full-blown desktops have reminded me how much simpler to deal with are laptops than desktops. Several months ago I set up a used computer from a friend in my office. I had little difficulty setting it up for the gentleman, and it ran so well that when a computer of identical description became available a bit later, I decided to buy it.

Big mistake. This second computer proved hopelessly buggy, crashing for no apparent reason with my standard suite of software. After a few weeks the floppy drives started to give up the ghost, memory problems got worse—I had a lemon on my hands. Changing the drive controller miraculously revived the floppies, but at a cost—no hard drives. Finally, convinced I didn't know what I was doing, I called someone who might. Half a day later: all drives functioning, but the memory problems were still there. God, how I hated that computer! Two more long days and nights and I had had enough; the last thing I could think of that I hadn't yet tried was to start swapping memory chips. If that didn't work, then I was just going to have to admit defeat, something I don't do easily. Out come a few chips, reinsert, reboot, and—lo and behold!—a machine with 8 MB of RAM was now telling me it couldn't access any more than 64 kB, not even enough to boot ROM-based diagnostics. More memory chip swapping, no progress. A phone call to a vendor brought the theory that I must have zapped all memory chips with static. Worriedly I remembered all the times I'd touched those chips without grounding myself first (how many of us really ground our computers, or anything?). Still, it didn't make sense. The drive controller was good, and I had been too careful handling the memory chips to zap them. It just HAD to be something else. Logic drew me next to the motherboard. Some of those other chips had to support the 8 MB of RAM—but which ones? Out of options, my mind searched for any possible solution...

The Woody Way: Once, years ago, Woody Hodgson told us about dealing with an ornery computer: "First thing is to push all chips that might conceivably be loose back in as far as they can go, and if that doesn't work, then clean the connectors." Well, it couldn't hurt to try that, methinks, so I started pressing all unsoldered chips into place, and every once in a while rebooting the machine to see if there had been any improvement. 7 or 8 slightly loose chips later, still only 64 kB on the 8 MB machine . Now, determining to strip the computer and resell the parts,while trying the last few chips... hmmm... a loose one there... Reboot, and: —Great Scott! (as Perry White of *Superman* used to say)—"8 MB of RAM available!" Ran my software, and it didn't crash; ran the highest-end software I could find in the office, and no crash. Months and months of wasted effort, long nights wasted when I could have been at home, a father and a husband. My real work lay piled up on my desk, some of it weeks behind. All this, and what for?— @*$!%&*!!

The Moral of the Story: When you've got computer problems you just can't understand, apply what I should have used at the very beginning —what is known in Tokyo as "The Hodgson Method" of computer repair. **Press in all chips which might conceivably be loose**, and see if that doesn't take care of things. Thanks, Woody!

● The recent talk of the town on TPC BBS and in last month's *TPC Newsletter* has been about the value—or lack thereof, judging by input from Steve Bellamy and David Parry—of shareware. I do agree that some shareware is garbage, but so is a lot of what finds its way into shrink-wrapped plastic. The only difference is in distribution method and—to me as a user and at times a large-scale buyer—that nice warm feeling I get from dealing with a company that trusts people to be honest.

I've spent more to test shrink-wrapped junk than I'd like my company to know about; but I've never bought shareware I was unhappy with. The most recent piece I will be shelling out for is the disk cache *Hyperdisk*, and its companion products: *Hyperkey* and a screen speedup routine.

I put *Hyperdisk* 4.2 to the test on my home machine, an original Compaq Deskpro 386 (old!), and on my Toshiba Dynabook 386SX. Test conditions were not optimum, but initial results, and the lack of problems so far, are very encouraging. I am not interested in the other 2 products, but *Hyperdisk* itself is outstanding. Just to get a feeling of what kind of differences are possible, look at the chart of *Norton SI* comparisons of my Compaq and Toshiba under differing cache conditions. The size of the cache makes a difference, but it isn't really noticeable under *SI*. When I run a database sort it will make a major difference, but 640 kB is enough for the kind of work I do on my Dynabook.

I am looking forward to testing this programwith a major (larger than 100 MB) database, to see whether it outperforms Compaq's old cache program. I don't use *SuperPCK* on any of my office machines for safety reasons. I found the old Compaq program perfectly safe, but *SuperPCK* sometimes trashed large databases. I hate reconstructing 50 MB+ files.

A test of a 640 kB *Hyperdisk* cache on a *Paradox* 725 kB Name & Address database on my Dynabook showed very good results, taking 22 seconds to run a complex sort for which the Compaq cache took 103 seconds. To be fair, the Compaq cache program came with a 1987 version of Compaq DOS.

WARNING: If you try *Hyperdisk* out on your computer, be sure to do *a full backup* first to avoid the risk of loss of work.

Roger Boisvert

	Computing Index	Disk Index	Performance Index
COMPAQ DESQPRO 386			
No cache program at all	13.1		
1 MB Compaq CACHE v. 1.2	17.6	13.3	16.1
1 MB SuperPCK	17.6	14.8	16.6
1 MB HYPERDISK	17.6	133.0	56.0
TOSHIBA DYNABOOK			
No cache program at all	15.3	4.8	11.1
640 kB Compaq CACHE v. 1.2	15.3	4.9	11.8
640 kB HYPERDISK	15.3	44.3	25.0

May Meeting Report

We are all newcomers once. This was the theme of forever-departing Steve Campbell's brief introductory message to the May meeting crowd, in which he described the humble underpinnings on which this now illustrious organization was established. After reportedly paying big bucks (i.e. $3,900) for an IBM PC back in 19 hundred and 83, he realized he needed technical assistance for formatting a floppy disk. The May meeting was intended, therefore, as a no-holds-barred Q & A session for people who still wonder why 5.25" floppies are floppier than 3.5" ones. Roughly half the audience indicated by a show of hands that they consider themselves newcomers to computing. Magnificent. Stuart Woodward, who was moderating the meeting, announced he would consider no query inane. As a result, a lot of basic and quite stimulating questions popped up.

One newcomer asked how he could make *WordPerfect* edit messages downloaded from an electronic bulletin board. This gave us enlightened ones, who suddenly found ourselves in the majority, a chance to point out the difference between word-processor files and "pure ASCII text" and how the one can be made the other. (A typical TPC discussion ensued over which is the Designated Key for performing this act on *WordPerfect*—read the hefty manual for the answer—and whether a plain ASCII editor wouldn't do the job better and faster.)

Another new voice asked for a brief explanation of "Windows", and Martin B., the club's *Windows* maven, gave it his best shot (it turns out *Windows* sits between DOS and the application software and demands that you play Solitaire when you are supposed to be working).

Someone, maybe not quite a neophyte, asked about back issues of PC Magazine and other computer-junkie stuff, and whether the material is stored on some gigantic electronic database. The answer turned out to be: try the PC-Magnet system for starters, which is accessed through CompuServe. It was further announced that the current big-shot editor of the Rag everyone loves to criticize is former TPC member Joel Dreyfuss, remembered fondly for his touting of shareware programs such as *ProComm, Scout* and *Qedit* which he downloaded fresh from New York when he was Fortune's Tokyo correspondent. Those members lucky to know Mr. Dreyfuss look forward to seeing his give-away spirit infect Ziff-Davis as well. Joel asked members for suggestions on how to improve his new magazine; we are sending him this ever-fresh publication for starters.

Other questions ranged from the ever-present dead batteries to dying or noisy power supplies, and how to make a hard disk bootable on a computer that insists on booting from a floppy (that problem stumped everyone present).

In the end, many of the neophytes admitted that, while the questions seemed fairly interesting, a lot of the replies washed over their heads. Well, we're all newcomers once, as Steve remarked. Now, if I could only get this floppy formatted in time .

THE **JUNE MEETING**

A TPC FIRST

PC'S AND MUSIC

demonstrating MIDI in action.

What is MIDI?
What are its benefits and limitations?
Who uses it?
And why?
How can you build a simple MIDI setup at home, using your PC?

@keyboard(Bruczkowski, Kazuko,Pianist,Concert)

in consort with

@dirige(@Sysop(Martin,"TPC Band"))

BE THERE!!

"Babbitoons"™© 1991 by A. Colville & D. Kach

Under Down Ar o und

and All About the (IBM Computer) World

Course Instructor: Steve Bellamy
Course Information: Eight Lessons @1.5 hours per lesson

LESSONS

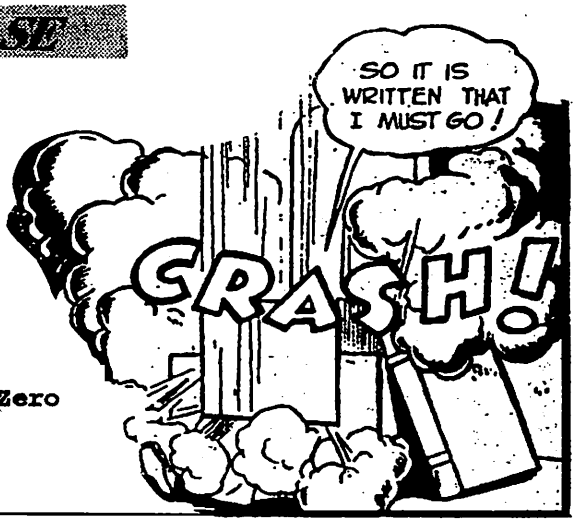
1 Hardware
2 The Operating System: DOS
3 The Network Extension: Ready, Steady, Connect
4 The Front End
5 All About Applications, Part I
6 All About Applications, Part II
7 Putting it all together and making it work right!
8 Productivity.
This class is a very special one, and there will be a prerequisite of attendance at four of the prior lessons. No exceptions, please!

Requirements
A notebook, a pencil, and an open mind. No previous experience is assumed. I will provide course notes, handouts, and other necessities.

The first lesson will be on Monday, 10 June, at 6:00 PM.
To enroll, contact Steve Bellamy at Mugen Computer Consultants, TEL/FAX (0473)78-4667. [Press 52 at the voice for FAX.]
Bruce Pierce will also answer your questions at (03)5567-1745.

TELL YOUR FRIENDS

CASE by CASE

DIVIDE OVERFLOW

```
Fatal Error:

Attempt to Divide by Zero
```